

Programmation Par Contraintes (1)

Exercice 1

On s'intéresse au problème suivant, posé initialement par Lewis Carroll :

Cinq maisons consécutives, de couleurs différentes, sont habitées par des hommes de différentes nationalités. Chacun possède un animal différent, a une boisson préférée différente et fume des cigarettes différentes. De plus, on sait que :

1. Le norvégien habite la première maison,
2. La maison à coté de celle du norvégien est bleue,
3. L'habitant de la troisième maison boit du lait,
4. L'anglais habite la maison rouge,
5. L'habitant de la maison verte boit du café,
6. L'habitant de la maison jaune fume des Kools,
7. La maison blanche se trouve juste après la verte,
8. L'espagnol a un chien,
9. L'ukrainien boit du thé,
10. Le japonais fume des cravens,
11. Le fumeur de old golds a un escargot,
12. Le fumeur de gitanes boit du vin,
13. Un voisin du fumeur de Chesterfields a un renard,
14. Un voisin du fumeur de Kools a un cheval.

=> À qui appartient le zèbre ?

Question 1 • Modélisez le problème du Zèbre de Lewis Carroll sous la forme d'un réseau de contraintes $N = \langle X, D, C \rangle$.

Question 2 • Complétez le programme MiniZinc.

Question 3 (bonus) • Implémentez sous Java une solution au problème et comparez les résultats avec le programme MiniZinc.

Exercice 2

Soit l'addition suivante :

$$\begin{array}{r} \text{SEND} \\ + \text{MORE} \\ \hline = \text{MONEY} \end{array}$$

Chaque lettre représente un chiffre compris entre 0 et 9. Nous souhaitons connaître la valeur de chaque lettre, sachant que la première lettre de chaque mot a une valeur différente de zéro.

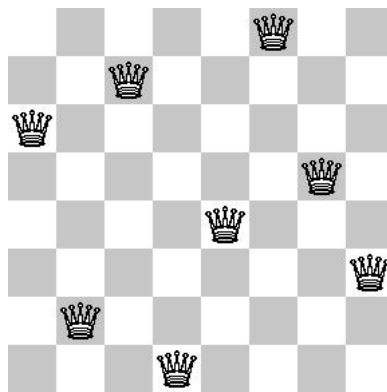
Question 1 • Modélisez le problème sous la forme d'un réseau de contraintes $N = \langle X, D, C \rangle$.

Question 2 • Utilisez la plateforme MiniZinc pour modéliser et résoudre le problème.

Question 3 (bonus) • Implémentez sous Java une solution au problème et comparez les résultats avec le programme MiniZinc.

Exercice 3

Soit un échiquier de $(N \times N)$ cases. Le problème des N -reines consiste à placer N reines de telle sorte qu'aucune reine ne puisse attaquer une autre.

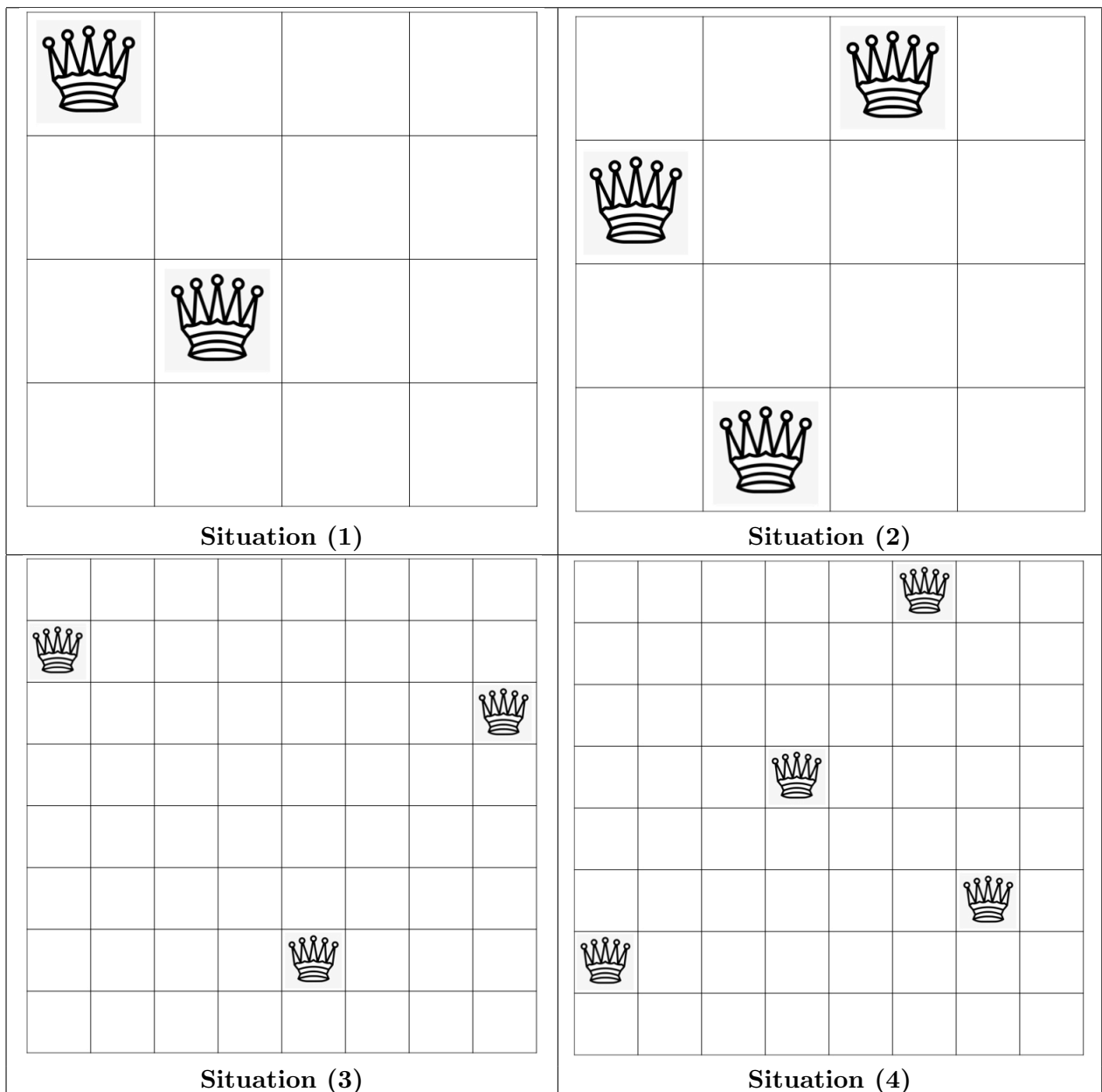


Question 1 • Modélisez le problème sous la forme d'un réseau de contraintes $N = \langle X, D, C \rangle$.

Question 2 • Déroulez l'algorithme du backtrack vu en cours sur l'instance (4×4) .

Question 3 • Utilisez maintenant la propagation et comparez la taille des deux arbres de recherche.

Question 4 • Les figures suivantes illustrent des instanciations partielles avec des valeurs affectées à quelques variables. La propagation des contraintes dans C permet de supprimer des valeurs de variables non instanciées. Donnez la liste des valeurs supprimées. Par exemple si la valeur 3 de la variable X_1 peut être supprimée en appliquant la contrainte C_2 , alors on notera dans la liste l'élément suivant : $C_2 \rightarrow (X_1, 3)$.



Question 5 • Utilisez la plateforme **MiniZinc** pour modéliser et résoudre le problème. Quelle est la taille de la plus grande instance qu'on peut résoudre avec le programme **MiniZinc** sous un contrat de temps d'une minute ?

Question 6 (bonus) • Implémentez sous Java une solution au problème et comparez les résultats avec le programme **MiniZinc**.

Exercice 4

Une règle de Golomb est une règle munie de marques à des positions entières telle que chaque paire de marques mesure une longueur différente.

Question 1 • Modélisez le problème sous la forme d'un réseau de contraintes $N = \langle X, D, C \rangle$.

Question 2 • Donnez la version optimisation du problème avec comme objectif de retourner la plus petite règle.

Question 3 • Utilisez la plateforme **MiniZinc** pour modéliser et résoudre le problème. Quelle est la taille de la plus grande instance qu'on peut résoudre avec le programme **MiniZinc** sous un contrat de temps d'une minute ?

Question 4 • Révisez votre modèle de sorte à résoudre une plus grande instance en une minute.

Question 5 (bonus) • Implémentez sous Java une solution au problème et comparez les résultats avec la PPC.

